Application of Metaheuristics for Solving Travelling Multitasker Problem and its Variants

A Synopsis

Submitted in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

in

Mathematics

Submitted By

Rani Kumari

Prof. Kamal Srivastava Supervisor Department of Mathematics	Prof. Ravinder Kumar Head, Department of Mathematics	Prof. G.S. Tyagi Dean, Faculty of Science

Department of Mathematics

Faculty of Science

Dayalbagh Educational Institute (Deemed University)

Dayalbagh, Agra-282110

May 2018

1. Introduction

Many of the optimization problems developed and studied over the past 50 years represented some real-world industrial or business environment. For some of these problems, exact optimal solutions are needed. Many of these problems are NP-hard, therefore exact methods consume large amount of time to obtain optimal solutions. To overcome this, heuristic approaches came into practice, with compromise on the optimality. Near optimal solutions are usually acceptable especially for large sized problem instances. But since these heuristics are problem specific, a more robust approach was required. This led to the use of metaheuristics to tackle the problems such as Traveling Salesman Problem (TSP), Knapsack Problem (KP), Packing while Traveling Problem, Capacitated arc routing Problem, Vehicle routing Problem etc.

The formal definition of metaheuristic is given as [8]:

"A metaheuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions."

The optimization problems mentioned above lack the complexity of most real-world problems. Large-scale business decision problems consist of interconnected components which need to be obtained, to achieve optimal solution for a business as a whole. Even if we know exact algorithm for subproblems, it remained an open question how to integrate these partial solutions to obtain a global optimum for the interconnected problem. Businesses need global solutions for their operations, not partial solutions. Earlier, reasons attributed to the hardness of a real-life problem were its size, constraints etc. But the complexity is not only due to these factors. As stated above the subproblems which constitute a real-world problem are interdependent to each other. These interdependences are responsible for the complexity of the real-world

problems, while this type of complexity in benchmark problems being currently in use is missing. Further, the benchmark problems used by researchers are same as 50 years ago. This is a reason behind growing gap between real-world problems and applicability of metaheuristic methods to real-world problems. Thus, a new class of benchmark problems that reflects the characteristic of real-world problems is needed.

As an illustration, one can consider the problem of transporting water tanks [21]. A company has a factory that produces water tanks of different sizes, and which are bundled inside each other to be loaded in trucks and transported to customers (stations). The cost involved is in its unbundling activity. Bundled tanks are unbundled at special sites (bases) before delivering them to customers. When the tanks are unbundled at a base only some of them would fit and are carried by the trucks, as they require more space. The remaining tanks are kept in the base until the truck gets back and loads them again to continue delivery process. The truck goes to each station, deliver the tanks, stops at some base to unbundle some tanks, takes subset of tanks to deliver. Truck repeats this activity until all tanks are delivered. The aim of optimizer is to select a subset of tanks for delivery and to determine exact route that also includes the bases in order to optimize the total value of delivery. Total value is proportional to the ratio between the total price of delivered tanks to the total distance that the truck travels. There are at least two subproblems embedded in this problem.

(i). Selecting the tanks for delivery/bundling (KP).

(ii). Delivering them to customers (TSP).

Clearly, an optimal solution to any one of these subproblems is not the solution of the given problem. For example, selection of tanks does not consider the location of bases. Another problem could be that of a delivery company which has to pick up packages at different locations which are combined in a route. For each collected package a fee is earned. But each

delivery van has only a certain weight capacity for cargo. Additionally, route costs are increased according to the weight of a package as the delivery van consumes more fuel and collecting a package decelerates the whole journey as the delivery man has to pick up the package at the client's location. Therefore, costs are increased through increase of fuel consumption, work time and other running costs of the delivery van. This problem again does not fit into any single optimization problem.

Observing these gaps, Bonyadi et.al. (2013) [21] introduced Traveling Thief Problem (TTP) by combining two existing NP hard problems, TSP and KP. In TTP a person called Thief carries a knapsack whose capacity is fixed. There are *n* cities and *m* items having some specified weights and profits. Thief has to travel these cities exactly once by picking some items from these cities in its knapsack with a constant knapsack capacity and has to return to the starting city. But by collecting items along the journey the weight of his knapsack increases and therefore decreases his velocity resulting in a longer total travel time. Also, the thief has to pay a rent for his knapsack which is proportional to the time taken in completing the tour. The objective of the thief is to find a tour such that time taken by tour is minimum and profit gained by knapsack's items is maximum.

The beauty of this optimization problem is that it tries to model the characteristics of real-world problems better by interconnecting two different problems with each other. Clearly in this case, the solution of one problem influences the other problem.

In order to define TTP formally, we first formally define TSP and KP.

Traveling Salesman Problem: TSP [2] is a classic NP-hard optimization problem. Let there are *n*-cities and distance between these cities are given by a matrix $D = \{d_{ij}\}$, where d_{ij} denotes distance between cities *i* and *j*. A salesman has to visit each city exactly once with constant velocity (v_c) and return to first city such that time/distance of tour is minimized. Let $x_i, i \in \{1, 2, \dots, n\}$, denotes ith city travelled by salesman.

Mathematically objective of Traveling Salesman Problem is to find a tour

$\overline{x} = (x_1, x_2, x_3, \dots, x_n)$, which

$$Minimize \ f(\overline{x}) = \sum_{i=1}^{n-1} \left(t_{x_i, x_{i+1}}\right) + \left(t_{x_n, x_1}\right)$$

$$\text{where } t_{x_i x_{i+1}} = \frac{d_{x_i x_{i+1}}}{v_c},$$

0-1 Knapsack Problem: In the theory of optimization this problem is categorized under a special class of problems, called packing problems. Assume that there are m number of items each with weight w_i, $i = 1, 2, 3, ..., m$ and profit p_i, $i = 1, 2, 3, ..., m$ and let the knapsack capacity be C. The objective is to maximize the total profit such that the item fits into the knapsack with respect to weight limit.

A solution to this problem can be represented as a vector $\overline{y} = (y_1, y_2, \dots, y_m), y_i \in \{0,1\} \ for \ i = 1,2, \dots, m,$ where $y_i = 1$ indicates that i^{th} item is included in the knapsack otherwise it is not present in the knapsack. Thus, mathematically knapsack problem is given by

$$Maximize \ g(\bar{y}) = \sum_{i=1}^{m} p_i y_i$$

$$Subject \ to \ \sum_{i=1}^{m} w_i y_i \le C$$

$$y_i \in \{0,1\}, i = 1, 2, ..., m$$

Interconnection: In the TTP these two subproblems get interconnected in a way that the solution of each subproblem is influencing the solution of the other subproblem. The more items the thief packs the more his velocity decreases and travel time increases. Let w_c and v be current weight and velocity respectively, which are given by $w_c = \sum_{i=1}^{c} y_i w_i$ and

$$v = v_{max} - \frac{w_c}{C}(v_{max} - v_{min})$$

where v_{max}, v_{min} are maximum and minimum velocity of thief respectively.

Thus, single objective TTP is mathematically defined as follows:

Single objective TTP: As the name suggests, in this problem a single objective needs to be optimized [21]. Let thief has to pay the rent. The knapsack rent rate is *R* rupees per unit time. Then objective of thief is to maximize total profit

$$\text{Maximize } G(\bar{x},\bar{y}) = g(\bar{y}) - Rf(\bar{x},\bar{y})$$
$$= \sum_{i=1}^{m} p_i y_i - R\left(\sum_{i=1}^{n-1}\left(t_{x_i,x_{i+1}}\right)+\left(t_{x_n,x_1}\right)\right)$$

$$\text{Subject to } \sum_{i=1}^{m} w_i y_i \le C$$

$$\text{where } t_{x_i,(x_{i+1})} = \frac{d_{x_i,(x_{i+1})}}{v_{x_i}}$$

$$v_{x_i} = v_{\max} - \frac{\sum_{j=1}^{i}\left(y(x_j)\cdot w_{x_j}\right)}{C}\left(v_{\max} - v_{\min}\right)$$

$$y_i \in \{0,1\}, i = 1,2,...,m$$

$$x \in B^n \text{ (Set of Permutation of n cities)}$$

Multi-objective TTP: Bonyadi et al. [21] also proposed multiobjectiveness into TTP. Mathematical model of multi-objective TTP is

$$\text{Maximize } g(\bar{y}) = \sum_{i=1}^{m} p_i y_i$$

$$\text{Minimize } f(\bar{x},\bar{y}) = \left(\sum_{i=1}^{n-1}\left(t_{x_i,x_{i+1}}\right)+\left(t_{x_n,x_1}\right)\right)$$

$$\text{Subject to } \sum_{i=1}^{m} w_i y_i \le C$$

$$\text{where } t_{x_i,(x_{i+1})} = \frac{d_{x_i,(x_{i+1})}}{v_{x_i}}$$

$$v_{x_i} = v_{\max} - \frac{\sum_{j=1}^{i}\left(y(x_j)\cdot w_{x_j}\right)}{C}\left(v_{\max} - v_{\min}\right)$$

$$y_i \in \{0,1\}, i = 1,2,...,m$$

$$x \in B^n \text{ (Set of Permutation of n cities)}$$

It is clear that the problem of maximization of $G(\bar{x},\bar{y}) = g(\bar{y}) - Rf(\bar{x},\bar{y})$ has been divided into two subproblems

(i). Max $g(\bar{y})$

(ii). Min $f(\bar{x},\bar{y})$

Paying of rent by thief is excluded here.

Many researchers have worked on TTP and also suggested some variants of TTP. We give here a brief survey of work done so far.

2. Metaheuristics

Metaheuristics can be classified as - population based and single solution based. In single solution-based metaheuristic, we start with a single initial solution and move away from it, describing a trajectory in the search space. Some of them can be seen as ‘‘intelligent’’ extensions of local search algorithms. Single solution-based metaheuristics are Simulated Annealing, Tabu Search, VNS etc. Population based metaheuristics deal with a population of solutions rather than with a single solution. A number of heuristics, local improvers, crossover, mutation operators etc. help the population to evolve towards better solutions. Normally bad solutions are eliminated from the population during the course of evolution and population converges eventually. Genetic Algorithms, Ant Colony Optimization, Scatter Search, NSGA II, etc. are some of the state-of-the-art population-based metaheuristics. Metaheuristics can be used to deal with single objective as well as multiobjective optimization problems. We briefly describe a few of them below:

2.1. Metaheuristics for single objective optimization problems

Variable Neighbourhood Search (VNS): VNS [7], is single solution-based metaheuristic and it uses various neighbourhood structures in the search space to find a single near optimal solution. For

VNS, we choose an initial solution *s* randomly or by some problem specific method and also its neighbour solution S'. Using local search, we find a local optimum S'' from this neighbourhood. If S'' is better than *s* (according to optimality) then only S'' is chosen as our new solution.

Simulated Annealing (SA): SA [7] is a local optimization method and is inspired by a model developed by Metropolis et al. for simulating the physical annealing process. This technique consists in carrying a material at high temperature, then in lowering this temperature slowly. The algorithm uses a pre-defined neighbourhood structure on the search space. A control parameter which is called ‘‘*temperature*’’ in analogy to the physical annealing process governs the search behaviour. In each iteration, a neighbour solution to the current solution is computed. If this neighbour solution has a better objective function value than the current solution, then the current solution is replaced by this neighbour solution. On the other hand, neighbour solution has a worse objective function value than the current one, then neighbour solution is only accepted with a certain probability depending on the difference of the objective function values in current and neighbour solutions, or the temperature parameter.

Genetic Algorithm (GA): GA [7] is a population based metaheuristic. Initially number of solutions (population) is generated by using some randomization process depending on problem. The basic GA is very generic, and there are many aspects that can be implemented differently according to the problem: representation of solution (chromosomes), selection strategy, type of crossover (the recombination operator of GA) and mutation operators, etc.

2.2. Metaheuristics for multi-objective optimization problems

Non-dominated sorting Genetic Algorithm (NSGA II): Deb in 2002 proposed NSGA II [11]. NSGA II is a metaheuristic for multiobjective optimization problem and it uses an explicit diversity preserving mechanism. In NSGA II offspring population is first created by using parent population. Non-dominated front is obtained using offspring and parent population both. Now elements of successive fronts (Starting with front one) become a member of parent

population for next generation, if cardinality of parent population of current generation is greater than or equal to sum of cardinality of front and parent population for next generation. If all member of front cannot become a member of parent population of next generation (cardinality of parent population is same in all generations), then crowding distance sorting is applied. The diversity among non-dominated solutions is introduced by using the crowding comparison procedure which is used in tournament selection and population reduction phase.

Strength Pareto Evolutionary Algorithms (SPEA): SPEA is an elitist evolutionary algorithm which is proposed by Zitzler and Thiele in 1998 [11]. This algorithm introduces elitism by explicitly maintaining an external population $\bar{P}$. This algorithm begins with a randomly created initial population P_0 of size N and empty external population $\bar{P}_0$ with maximum capacity of $\bar{N}$. In any generation *t*, the best non-dominated solution of population P_t are copied to the external population $\bar{P}_t$. The dominated solutions in the modified external population are found and deleted from the external population. In this way elites are preserved in external population.

2.3. Hybrid Algorithms

Hybrid Algorithms are combination of two or more metaheuristics or metaheuristic with local search. Percy et.al. [25] in 1995 proposed Guided Evolutionary Simulated Annealing (GESA) technique which combines GA and Simulated annealing.

Srivastava et.al. [12] in 2010 proposed Hybrid Meta-Heuristic (HMH) for multiobjective optimization problem, which combines GA and SA. In HMH initial population is considered as parents population. Each parent is allowed to produce children by crossing over with a randomly selected string from the remaining population of parents. Parents together with their children is referred to as a family and all solutions of this family are referred as family members and they are used to find pareto front (set of nondominated solutions). If no members of family

are considered in pareto front then all the members of this family are not rejected together in the hope of its improvement in future. A member of that family which is ‘nearest’ to the Pareto front is selected to be the parent for the next generation.

3. Literature Review

3.1. Single objective TTP

After the TTP was first introduced [21], it was studied by Polyakovskiy et.al. [28] (2014). They proposed a set of two-step heuristics for solving single objective TTP. In the first step they used Lin-Kernighan heuristic [2] to find a good tour for TSP part of TTP without considering KP part of the problem and fixed it for some duration to find good solution of TTP. In the second step they proposed three packing plan heuristics, namely Simple heuristic (SH), Random Local Search (RLS) and a simple (1+1) Evolutionary Algorithm (EA). SH is a constructive heuristic in which after generating tour for TSP, the score value (total profit that the thief gets if and only if item is picked) is calculated for each item to estimate how good it is according to present tour. After calculating score value, item is picked one after another in descending order of score value. RLS and EA work on best-so-far, in which they record the best solution so far and then repeatedly create a new single solution based on some probabilistic calculations. Solution is compared against the best-so-far, if TTP objective value of solution is better than best-so-far then this new single solution is considered as best-so-far, otherwise it is ignored.

Bonyadi et.al. [20], (2014) gave two heuristic approaches, Density Based (DH) and CoSolver, which is a Socially Inspired Algorithm for solving single objective TTP. DH uses Chained-Lin-Kernighan to generate a tour and then they try to find the best picking plan for the obtained tour. CoSolver decomposes TTP into sub-problems, and then solves the sub-problems separately with some communication between them and then composes the solutions back to

obtain a solution of TTP. They finally conclude that considering interdependency is beneficial in solving multi-component problems.

Mei. et.al. [31], (2014) proposed two meta-heuristics for TTP. Cooperative Co-evolution (CC) which solves subproblems separately and transfers the information between them in each generation. The other one is Memetic Algorithm (MA) for the overall problem. In MA TSP and KP are solved together by combining the aforementioned operators.

Mei. et.al. [32], (2014) analysed complexity of large scale single objective TTP. They proposed memetic algorithm with two stage local search for complexity reduction. They solved TTP with more than 10,000 cities and 100,000 items within limited computational budget.

Faulkner et.al. [4], (2015) proposed a number of packing operators (Packing Routine, Bitflip, Insertion) which run on good TSP tour derived by Chained Lin-Kernighan heuristics [2]. They compared results with an approximate mixed integer programming-based approach and existing heuristic solution.

Mei et.al. [30], (2015) gave Two-Stage memetic algorithm (TSMA) for single objective TTP. TSMA is combination of genetic algorithm and local search. Genetic Programming (GP) is adopted to evolve effective Item Selection Heuristic. Two heuristics GAIN and PICKFUNC heuristics are designed to replace different components of the original item selection heuristics.

Yafrani and Ahiod [14], (2016) proposed two algorithms for TTP, a single solution based and a population based memetic algorithm. They found that both algorithms give competitive performance on different TTP instances.

An ant colony optimization algorithm for TTP was proposed by Wagner [23], (2016) in which the focus is on generating good TTP tours using ants instead of going for profitable packing plans on good TSP tours.

Wagner et.al. [24], (2017) contributed to TTP in four ways. They created a comprehensive data set that comprises of performance data of 21 TTP algorithms on the full original set of 9720 TTP instances. They defined 66 characteristics for all TTP instances that can be used to select the best algorithm on a per-instances basis. They used these algorithms and features to construct the first algorithm port-folios for TTP, clearly outperforming the single best algorithm. Finally, they studied as to which algorithm contribute most to this portfolio.

Moeini et.al. [19], (2017) introduced hybrid algorithm for single objective TTP, that combined GA, local search (LS) and Variable Neighbourhood Search (VNS).

Martins et al. [22], (2017) and Yafrani et al. [16], (2018) proposed hyper heuristic approach for TTP and their hyper heuristics are based on estimation of distribution of algorithm and low-level heuristics respectively.

Recently Yafrani and Ahiod [15], (2018) have given two algorithms based on a 2-OPT steepest ascent hill climbing algorithm and the simulated annealing metaheuristic named CS2SA and CS2SA-R respectively to solve the problem. The obtained results show that the proposed algorithms are efficient for many TTP instances of different sizes and properties.

Araujo et.al. [26], (2018) have proposed a novel list-constrained local search process inspired by Variable Neighbourhood Descent (VND) for multiple neighbourhood structures, combined with a metaheuristic Greedy Randomized Adaptive Search Procedure (GRASP). They have implemented local search in a Graphics Processing Unit (GPU).

3.2. Multi-objective TTP

Blank et.al. [9], (2017) gave three algorithms: greedy Algorithm, Independent Subproblem Algorithm and NSGAII to solve Bi-Objective Traveling thief problem. They compared their results obtained by each algorithm by finding hypervolume of normalised median pareto front and concluded that NSGAII gives best results. There is still a lot of scope to design better greedy

algorithms, local improvement heuristics, solution construction heuristics which may lead to improvement in the search process.

Yafrani et.al. [17], (2017) studied multi-objectiveness in the Single-objective traveling thief problem. Their proposed algorithm is built around the NSGA-II framework. Their results indicate that that a multiobjective approach can produce solutions to the single objective TTP variant while being competitive to current state-of-art solvers.

3.3. Variants of TTP

Chand and Wagner [27], (2016) presented as reaction to criticism that the TTP is not realistic and they introduced multiple traveling thieves problem as a variation of original traveling thief problem. They suggested that if there is a group of thieves then they can gain more profit in less travel time. In MTTP it is not necessary to visit all cities by a single thief. Thieves may divide the cities amongst themselves to visit. Two thieves can visit same city but they are not to collect same items from same city. Starting and ending city of all thieves is same, so that they can distribute loot material. Total profit from loot is sum of profits gained by each thief. Objective of MTTP is to maximize total profit gained by loot. For MTTP one has to simultaneously optimize several tours and packing plans. It is pertinent to note here that metaheuristic-based solution strategies are not available in literature for these recently introduced problems.

Junhua et.al. [10], (2018) introduced a new bi-objective version of TTP. They introduced an additional objective in the original single objective TTP (BO-TTP) wherein the total accumulated weight is to be minimized. They proposed a hybrid indicator-based evolutionary algorithm that utilises the exact dynamic programming algorithm for the underlying packing while traveling problem as a subroutine to evolve the individuals.

As mentioned earlier, the motivation for introducing TTP has been the lack of complexity in the prevalent optimization problems. The 0-1 KP was not sufficient to model many similar problems [3], [6] and [13] which led to defining other variants of KP. Some of the variants of KP which exist in literature and are relevant to or are motivated by real-world problems are briefly described below.

Multiple Knapsack Problem (MKP) [18]: Simple knapsack problem was introduced with only one knapsack but a person may want to fill multiple knapsacks (say k knapsacks) with varying capacities. Let capacity of *i*th knapsack is C_i. A solution to the problem is represented by a matrix ($\bar{y}$) of order $m \times k$, where $y_{ij} \in \{0,1\}$, $y_{ij} = 1$ if *j*th knapsack contain *i*th items.

Multiple knapsack problem is defined mathematically as

$$Maximize \quad f(\bar{y}) = \sum_{i=1}^{m} \sum_{j=1}^{k} p_i y_{ij}$$

$$Subject \ to \quad \sum_{i}^{m} w_i y_{ij} \le C_j \quad j = 1, 2, \ldots, k$$

$$\sum_{i=1}^{m} y_{ij} \le 1, \quad j = 1, 2, \ldots, k$$

$$y_{ij} \in \{0,1\} \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, k$$

Multi-dimensional Knapsack Problem [6]: Assume that we have to choose certain number (*L*) of projects from set of *m* projects such that their combined utility is maximized. Combined utility of projects is sum of individual projects utility. Let there are k different resources available for these projects, v_i be the fixed utility of i^{th} project for $i = 1, 2, \ldots, m$, w_{ij} be a fix amount of j resource available for i^{th} project for $j \in \{1,2, \ldots, k\}$, $i \in \{1,2, \ldots, m\}$ and W_j be the limited amount of j^{th} resources. The Multi-dimensional knapsack problem for selection of projects is formulated as

$$\text{Maximize } g(\bar{y}) = \sum_{i=1}^{m} v_i y_i$$

$$\text{Subject to } \sum_{i=1}^{m} w_{ij} y_i \le W_j \ , \ j = 1, 2, ..., k$$

$$\sum_{i=1}^{m} y_i \le L$$

$$y_i \in \{0,1\}, i = 1, 2, ..., m$$

where $\bar{y}$ is same as define in simple 0-1 knapsack problem.

Quadratic Knapsack Problem [3]: Quadratic knapsack problem was introduced by Gallo et.al. [3] in 1976. Let Q is a non-negative square matrix of order m, $a = (a_1, a_2, ..., a_m)$ is a positive m-vector and b a positive scaler. Since $y_i = y_i^2$ for $y_i = 0, 1$ linear terms in the objective function can be implicitly taken into account in the quadratic part. Mathematically Quadratics Knapsack problem is given by

$$\text{Maximize } g(\bar{y}) = \bar{y}^T Q \bar{y}$$

$$\text{Subject to } \sum_{i=1}^{m} a_i y_i \le b$$

$$y_i \in \{0,1\}, i = 1, 2, ..., m$$

where $\bar{y}$ is same as defined in simple 0-1 knapsack problem

Multiobjective Knapsack Problem [1]: In Multiobjective Knapsack problem we have to optimize more than one objective functions (requirements) at same time. Let $g_1(y), g_2(y), ..., g_K(y)$ be K objective functions. Mathematical model of Multiobjective Knapsack Problem is

$$\textit{Maximize } g_k(\bar{y}) = \sum_{i=1}^{m} c_i^k y_i \qquad k = 1, 2, ..., K$$

$$\textit{Subject to} \quad \sum_{i=1}^{m} w_i y_i \le C$$

$$y_i \in \{0,1\}$$

Bi-Level Knapsack Problem [13]: In Bi-level knapsack problem, we have two knapsacks which may be of different capacities carried by two different investors, leader (government) and follower (Company). Leader has to optimize profit carried out by both investor. But Follower bothers only about his own profit. If ith item is carried out by both investors then both will gain profit $p_i + a_i$, where a_i is any positive quantity. Let $\bar{y}^{(1)}$ and $\bar{y}^{(2)}$ are binary vectors containing packing plan of leader and follower respectively. C_1 and C_2 are capacities of knapsack carried by leader and follower respectively. Mathematical model of Bi-level Knapsack Problem is

$$\textit{Maximize } \mathrm{g}_1\left(\bar{y}^{(1)}, \bar{y}^{(2)}\right) = \sum_{i=1}^{m}\left(y_i^{(1)} + y_i^{(2)}\right) p_i + 2\sum_{i=1}^{m} a_i y_i^{(1)} y_i^{(2)}$$

$$\textit{Maximize} \quad \mathrm{g}_1\left(\bar{y}^{(1)}, \bar{y}^{(2)}\right) = \sum_{i=1}^{m} y_i^{(2)} p_i + \sum_{i=1}^{m} a_i y_i^{(1)} y_i^{(2)}$$

$$\textit{Subject to} \quad \sum_{i=1}^{m} w_i \bar{y}_i^{(1)} \le C_1$$

$$\sum_{i=1}^{m} w_i \bar{y}_i^{(2)} \le C_2$$

$$\bar{y}_i^{(1)}, \quad \bar{y}_i^{(2)} \in \{0,1\}, \quad i = 1, 2, ..., m$$

Fractional Knapsack Problem [5]: In the 0-1 Knapsack problem, we are not allowed to break items. We either take the whole item or don't take it. But in **fractional knapsack problem**, we can break items for maximizing the total value of knapsack.

4. Objectives

TTP reduces some gap between real world and benchmark problems, using two subproblems. But there are many other open real-world problems which can be modelled by the hybridization of variants of Knapsack and TSP problems. In light of the work available in the literature for TTP and the gaps and open issues discussed in Section 3, we list the main objectives of this work.

1. To design better metaheuristic based approaches to solve TTP.
2. Develop metaheuristics for the Multiobjective TTP and Multiple traveling thief problem.
3. To evaluate the performance of the developed metaheuristic approaches on standard benchmark instances.
4. Perform complexity analysis of developed heuristics/algorithms.
5. Study the possible combination of KP and TSP variants with reference to real life problems.
6. To explore new objectives which may be added to the existing set of objectives of TTP depending upon the real-life problems.

References

[1]. C. Bazgan, H. Hugot and D. Vanderpooten. *Solving efficiently the 0–1 multi-objective knapsack problem.* Computers and Operations Research, Science Direct, Elsevier, Vol. 36, pp 260-279, 2009

[2]. D. Applegate, W. J. Cook and A. Rohe. *Chained Lin-Kernighan for large travelling salesman problems*. INFORMS Journal on Computing, 15, pp. 82-92, 2003.

[3]. G. Gallo, P. L. Hammer and B. Simeone, *Quadratic Knapsack Problems*. Mathematical Programming, Vol. 12, pp 132-149, 1980.

[4]. H. Faulkner, S. Polyakovskiy, T. Schultz and M. Wagner. *Approximate approaches to the traveling thief problem*. GECCO, pp 385-392, ACM, 2015.

[5]. H. Ishii, T. Ibaraki and H. Mine. *Fractional Knapsack Problems*. Mathematical Programming Vol. 13, pp 255-271, 1977

[6]. H. M. Weingartner and D. N. Ness. *Methods for the solution of the Multi Dimensional 0/1 Knapsack Problem,* Operations Research, Vol. 15, No. 1, 2015.

[7]. I. Boussaid, J. Lapagnot and P. Siarry. *A survey on optimization metaheuristic,* Information Sciences. Vol. 237, pp 82-117, 2013.

[8]. I. H. Osman and G. Laporte. *Metaheuristic: A bibliography*. Annals of Operation Research, Vol. 63, pp 513-623, 1996.

[9]. J. Blank, K. Deb and S. Mostaghim. *Solving the Bi-objective Traveling Thief Problem with Multi-objective Evolutionary Algorithms.* Springer International Publishing House, EMO 2017, LNCS 10173, pp. 46-60, 2017.

[10]. J. Wu, S Polyakovskiy, M Wagner and F Neumann, *Evolutionary Computation plus Dynamic Programming for the Bi-Objective Travelling Thief Problem,* Accepted GECCO 2018.

[11]. K. Deb, *Multi Objective Optimization Using Evolutionary Algorithms*, Publisher- John Wiley & Sons LTD, 2008.

[12]. K. Srivastava, S. Srivastava, B. K. Pathak and K. Deb, *Discrete Time-Cost Tradeoff with a Novel Hybrid Meta-Heuristics,* Lecture Notes in Economic and Mathematical Systems Vol. 634, 2010.

[13]. L. Chen and G. Zhang. *Approximation algorithms for a bi-level knapsack problem,* Theoretical Computer Science, Elsevier, Vol. 497, pp: 1-12, 2013.

[14]. M. E. Yafrani and B. Ahiod. *Population -based vs Single solution Heuristics for the travelling thief problem*. GECCO 2016.

[15]. M. E. Yafrani and B. Ahiod, *Efficiently Solving the Traveling Thief Problem using Hill Climbing and Simulated Annealing*, Information Sciences, Vol. 432, pp: 231-233 2018.

[16]. M. E. Yafrani, M. Martins, M. Wagner, B. Ahiod, M. Delgado and R. Luders. *A Hyperheuristic approach based on low-level heuristics for the travelling thief problem*. Genetic Programming and Evolvable Machines, Vol. 19, No. 1-1, pp: 121-150, Springer 2018.

[17]. M. E. Yafrani, S. Chand, A Neumann, B. Ahiod and M Wagner. *Multi-objectiveness in the Single-objective Traveling Thief Problem.* GECCO, 2017.

[18]. M. E. Lalami, M. Elkihel, D. E. Baz and V. Boyer. *A procedure-based heuristic for 0-1 Multiple Knapsack Problems.* Int. J. Mathematics in Operational Research, Vol. 4, No. 3, 2012.

[19]. M. Moeini, D. Schermer and O. Wendt. *A hybrid Evolutionary Approach for solving the Traveling thief Problem,* International Conference on computational science and its application, Springer, pp 652-668, 2017

[20]. M. R. Bonyadi, Z. Michalewicz, M. R. Przybylek and A. Wierzbicki. *Socially inspired algorithms for the TTP*. GECCO 2014, pp. 421-428, ACM, 2014.

[21]. M. R. Bonyadi, Z Michalewicz and L Barone. *The traveling thief problem: the first step in the transition from theoretical problem to realistic problems.* IEEE Congress on Evolutionary Computation, pp. 1037-1044, 2013.

[22]. M. S. R. Martins, M. E. Yafrani, M. R. B. S. Delgado, M. Wagner, B. Ahiod and R. Luders. *HSEDA: A Heuristic Selection Approach Based on Estimation of Distribution Algorithm for the Travelling Thief Problem.* GECCO 2017.

[23]. M. Wagner. *Stealing Items More Efficiently with Ants: A Swarm Intelligence Approach to the Travelling Thief Problem*. Dorigo M. et.al. (eds) Swarm Intelligence. ANTS 2016. Lecture Notes in Computer Science, vol. 982. Springer, Cham 2016.

[24]. M. Wagner, M. Lindauer, M. Misir, S. Nallaperuma and F. Hutter. *A case study of algorithm selection for the traveling thief problem*. Journal of Heuristics, Springer, pp. 1-26, 2017.

[25]. P. C. Yip, Y. H. Pao, *Combinatorial Optimization with Use of Guided Evolutionary Simulated Annealing,* IEEE Transactions on Neural Networks, VOL. 6. NO. 2, 1995.

[26]. R. P. Araujo, E. Rios, I. M. Coelho, L. A. J. Marzulo and M. C. Castro, *A novel List-Constrained Randomized VND approach in GPU for the Traveling Thief Problem.* Electronic notes in Discreate Mathematics, Science Direct, Elsevier, Vol. 66, pp 183-190, 2018.

[27]. S. Chand and M. Wagner. *Fast Heuristics for the Multiple Traveling Thieves Problem*. GECCO, 2016.

[28]. S. Polyakovskiy, M. R. Bonyadi, M. Wagner, Z. Michalewicz and F. Neumann. *A Comprehensive benchmark set and heuristics for the travelling thief problem.* GECCO, pp. 477-484, ACM, 2014.

[29]. W. Shao, F. D. Salim and T. G. Dinh. *Travelling Officer Problem: Managing Car Parking Violations Efficiently Using Sensor Data.* IEEE Internet of Things, 2018

[30]. Y. Mei, X. Li, F. Salim and X. Yao. *Heuristic Evolution with Genetic Programming for Traveling Thief Problem.* IEEE. 2015

[31]. Y. Mei., X. Li, and X. Yao. *On investigation of interdependence between subproblem of the travelling thief problem.* Soft Computing, 20(1), pp. 157-172, 2014.

[32]. Y. Mei, X Li and X yao, *Improving Efficiency of Heuristics for the large scale traveling thief problem.* International publishing switzerland, pp. 631-643, 2014.